Kelompoak 1

**Studi Kasus: Memilih Mata Kuliah Semester Depan**

**Pemahaman Masalah:**

Seorang mahasiswa ingin memilih mata kuliah untuk semester depan. Setiap mata kuliah memiliki bobot (SKS) dan nilai manfaat (misalnya, minat, relevansi dengan karir). Mahasiswa ingin memaksimalkan nilai manfaat total tanpa melebihi beban studi maksimum yang diizinkan.

**Pemodelan dengan Knapsack:**

* **Item:** Setiap mata kuliah adalah sebuah item.
* **Bobot:** SKS dari mata kuliah.
* **Nilai:** Nilai manfaat dari mata kuliah.
* **Kapasitas Knapsack:** Beban studi maksimum yang diizinkan.

**Tujuan:** Memilih kombinasi mata kuliah yang memaksimalkan nilai total tanpa melebihi beban studi maksimum.

**Algoritma Knapsack 0/1:**

Algoritma ini cocok untuk masalah di mana kita hanya bisa memilih satu item sekali (mengambil atau tidak mengambil mata kuliah). sambil menambahkan fitur tracing back untuk melihat kombinasi mata kuliah yang dipilih.

def knapsack(values, weights, capacity):

n = len(values)

dp = [[0 for x in range(capacity + 1)] for x in range(n + 1)]

# Inisialisasi matriks untuk menyimpan item yang dipilih

keep = [[False for x in range(capacity + 1)] for x in range(n + 1)]

for i in range(1, n + 1):

for w in range(1, capacity + 1):

if weights[i-1] <= w:

if values[i-1] + dp[i-1][w-weights[i-1]] > dp[i-1][w]:

dp[i][w] = values[i-1] + dp[i-1][w-weights[i-1]]

keep[i][w] = True

else:

dp[i][w] = dp[i-1][w]

else:

dp[i][w] = dp[i-1][w]

# Fungsi untuk melakukan tracing back

def traceback(i, w):

if i == 0 or w == 0:

return

if keep[i][w]:

traceback(i-1, w-weights[i-1])

print(f"Pilih mata kuliah {i} dengan nilai {values[i-1]}")

else:

traceback(i-1, w)

traceback(n, capacity)

return dp[n][capacity]

# Contoh penggunaan:

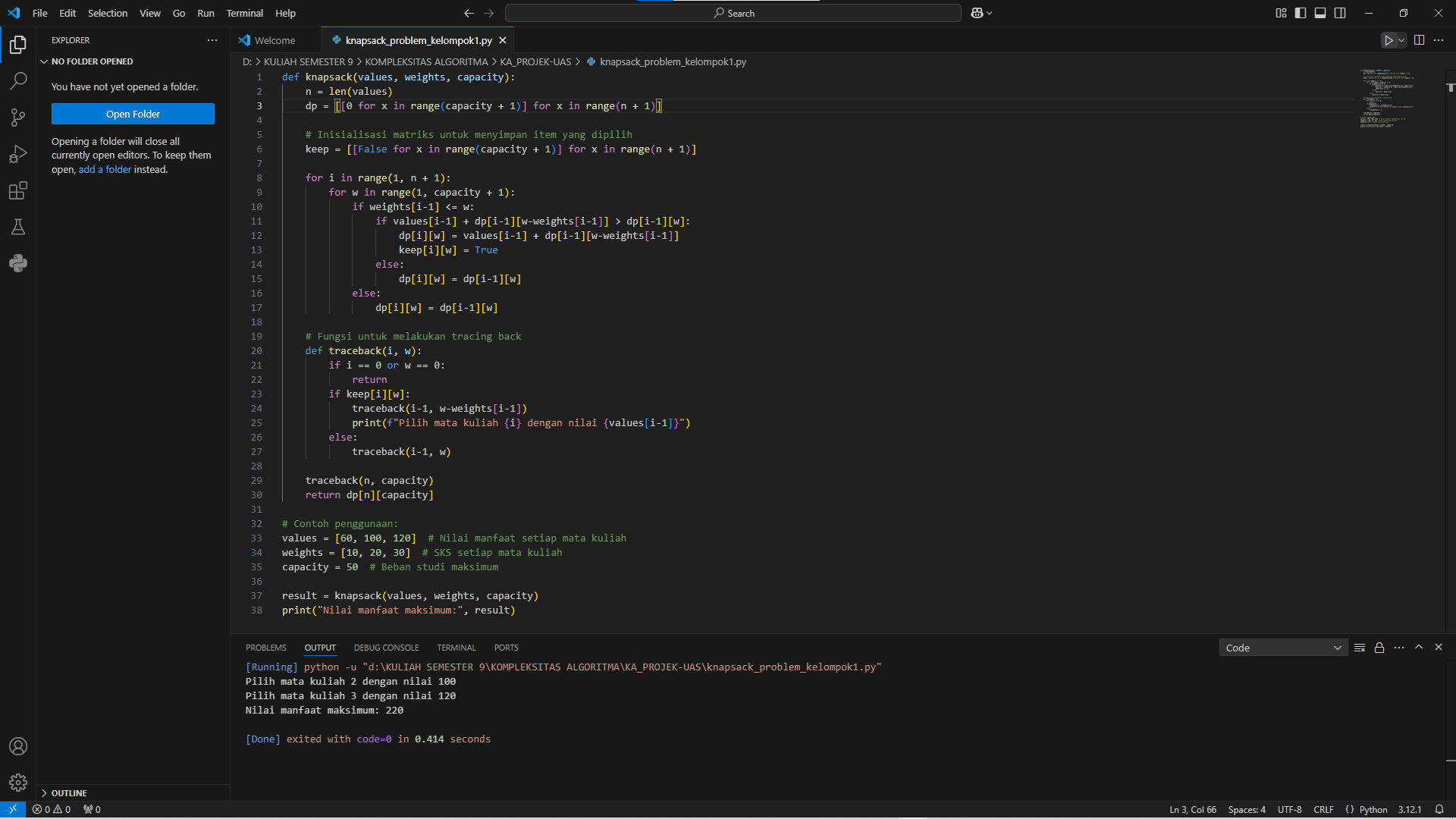
values = [60, 100, 120] # Nilai manfaat setiap mata kuliah

weights = [10, 20, 30] # SKS setiap mata kuliah

capacity = 50 # Beban studi maksimum

result = knapsack(values, weights, capacity)

print("Nilai manfaat maksimum:", result)



**Penjelasan:**

1. **Matriks keep:** Matriks ini digunakan untuk mencatat apakah suatu item (mata kuliah) dipilih atau tidak dalam solusi optimal.
2. **Fungsi traceback:** Fungsi ini secara rekursif melacak kembali keputusan yang diambil dalam mengisi matriks dp untuk menentukan kombinasi item yang dipilih.
3. **Output:** Selain menampilkan nilai manfaat maksimum, program juga akan mencetak indeks mata kuliah yang dipilih.

**Cara Kerja:**

* **Pengisian dp dan keep:** Saat mengisi matriks dp, kita juga mengisi matriks keep dengan True jika suatu item dipilih untuk mencapai nilai maksimum pada sel tersebut.
* **Tracing Back:** Fungsi traceback dimulai dari sel kanan bawah matriks dp (nilai maksimum yang ditemukan). Jika nilai pada sel tersebut sama dengan sel di atasnya, berarti item saat itu tidak dipilih. Jika berbeda, berarti item tersebut dipilih, dan kita rekursif memanggil fungsi traceback untuk sel sebelumnya dengan kapasitas yang dikurangi bobot item yang dipilih.